EXHIBIT AA

TO DECLARATION OF S. MERRILL WEISS IN SUPPORT OF PLAINTIFF ACACIA MEDIA TECHNOLOGIES CORPORATION'S MEMORANDUM OF POINTS AND AUTHORITIES IN OPPOSITION TO ROUND 3 DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY UNDER 35 U.S.C. § 112 OF THE '992, '863, AND '702 PATENTS; AND SATELLITE DEFENDANTS' MOTION FOR SUMMARY JUDGMENT OF INVALIDITY OF THE '992, '863, AND '720 PATENTS

Hovsepian v. Apple, Inc.

Doc. 325 Att. 28

Dockets.Justia.com

Brief Papers

Derivation of a Class of Training Algorithms

S. P. LUTTRELL

Abstract—**This paper presents a novel derivation of Kohonen's topographic mapping training algorithm, based upon an extension of the Linde-Buzo-Gray (LBG) algorithm for vector quantizer design. Thus a vector quantizer is designed by minimizing an L_2 reconstruction distortion measure, including an additional contribution from the effect of code noise which corrupts the output of the vector quantizer. The neighborhood updating scheme of Kohonen's topographic mapping training algorithm emerges as a special case of this code noise model. This formulation of Kohonen's algorithm is a specific instance of the "robust hidden layer principle," which stabilizes the internal representations chosen by a network against anticipated noise or distortion processes.**

I. Introduction

Vector quantization theory is a generalization of scalar quantization theory as expressed by the Lloyd–Max equations [1], [2]. The generalization of the Lloyd–Max equations to vector quantization is straightforward, and the Linde–Buzo–Gray (LBG), or k means, algorithm [3] provides a means of adjusting the code vectors to locate a local minimum of a distortion measure.

In this paper, the L_2 distortion measure is extended to include the effect of corruption of the vector quantizer output code. Kohonen's topographic mapping training algorithm then emerges naturally. Some extensions of the technique are indicated.

II. The LBG Algorithm [3]

Although it is not strictly necessary, we shall restrict our attention to an L_2 distortion measure $d(x, x')$ defined as the following, which measures the Euclidean distance between a vector x and its reconstruction after vector quantization x':

$$d(x, x') \equiv [x' - x]^2. \quad (1)$$

We shall be concerned with the average of $d(x, x')$ over a training set of vectors x, so we shall introduce the probability density function (PDF) $P(x)$ over samples x selected at random from the training set.

Denote the encoding operation as $y(x)$, and the corresponding decoding operation as $x'(y)$. The *average* L_2 distortion D is the average of $d(x, x'(y(x)))$ over x sampled from $P(x)$

$$D \equiv \int dx\, P(x)[x'(y(x)) - x]^2. \quad (2)$$

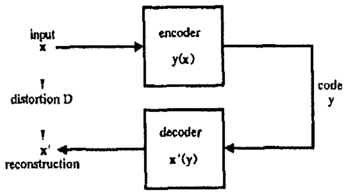

Fig. 1. Representation of (2) describing the L_2 distortion D after encoding using $y(x)$ and then decoding using $x'(y)$. This is a single stage encoder.

Equation (2) is depicted schematically in Fig. 1, where $x'(y)$ acts as a pseudoinverse for $y(x)$. The optimum encoding/decoding scheme is found by appropriately varying the functions $y(x)$ and $x'(y)$ so as to minimize D. Necessary conditions for minimization of D are [3]:

A1) Given x, the appropriate code $y = y(x)$ minimizes the distortion $[x'(y) - x]^2$.

A2) given y, the appropriate reconstruction $x' = x'(y)$ is the centroid of those x's which satisfy $y = y(x)$.

Note that A1 is a nearest neighbor encoding rule. A1 and A2 imply that D is stationary (and locally minimum) with respect to variations of $y(x)$ and $x'(y)$, respectively. Such conditions are therefore sufficient for a local minimum of D, but are only necessary for a global minimum. The LBG algorithm runs in batch training mode where the whole training set is presented before performing an update, which consists of alternately adjusting $y(x)$ according to A1, and then adjusting $x'(y)$ according to A2.

III. Two Stage Quantization

Now consider a scheme in which the encoding operation is broken down into two stages, $h(x)$ followed by $y(h)$, where h is an intermediate code. The L_2 distortion is then given by

$$D \equiv \int dx\, P(x)\left[x'\left(y\left(h(x)\right)\right) - x\right]^2. \quad (3)$$

Equation (3) is more clumsy than (2) because there are many ways of achieving the same overall encoding oper-

Manuscript received June 2, 1989; revised November 27, 1989. An earlier version of this paper was presented at the 1989 International Conference on Neural Networks, Washington, DC, June 19–22, 1989.
The author is with the Royal Signals and Radar Establishment, Malvern, Worcs., WR14 3PS, U. K.
IEEE Log Number 9034406.

1045-9227/90/0600-0229$01.00 © 1990 British Crown Copyright

Exhibit AA Page 511

ation $y(h(x))$ by delicately balancing the functional forms of $h(x)$ and $y(h)$. In general, the functional forms of $h(x)$ and $y(h)$ are not simple, although $y(h(x))$ itself is a vector quantizer.

The essential step is to model the *average* distortion due to the second stage of encoding $y(h)$ as a noise process acting on the output of the first stage $h(x)$. Thus, consider a modified distortion defined as

$$D_1 \equiv \int dx\, P(x) \int dn\, \pi(n)\left[x'(h(x) + n) - x\right]^2. \tag{4}$$

Equation (4) is depicted in Fig. 2. This is the L_2 distortion which would arise from encoding x using $h(x)$, followed by addition of noise n with PDF $\pi(n)$ (to produce h'), followed by decoding using $x'(h')$. By a suitable choice of $\pi(n)$ we can model the *average* distortion due to $y(h)$. Note that we use a somewhat cavalier notation by using $x'(h')$ here and $x'(y)$ elsewhere: this does not imply that the functions are the same.

The use of a modified distortion measure (D_1 rather than D) is a specific instance of the application of the "robust hidden layer principle." This principle states that a network should be trained in the presence of all anticipated noise and distortion processes acting on its internal representations. The effect of this is to encourage the network to encode information in its internal representations in such a way that it is robust with respect to the damaging effects of noise and distortion. In this case we use $\pi(n)$ to model stochastically the effects of the distortion process.

The noise model that we have used in (4) is additive. We could use more sophisticated noise models to take into account the different distortions introduced by $y(h)$ for different h, in which case $\pi(n)$ would become $\pi(n \mid h)$. We do not study such an extension of the model in this paper.

In Fig. 3 we show the set theoretic version of Fig. 2. Note how the distortion is intuitively and consistently represented by the sizes of the subsets depicted. Neither $h(x)$ nor $y(h)$ is guaranteed to have a simple form (only $y(h(x))$) is a complete vector quantizer), but we have nevertheless represented them as if they mapped clusters of nearby points to the same output: this is purely for diagrammatic convenience. The equivalent noise which $y(h)$ induces on a particular h (h_0 say) is then represented by the set of h which satisfy $y(h) = y(h_0)$. In Fig. 3 we show how four separate h values are mapped to the same y value, and we show how each of those four h values derives from a small cluster of x values, and how the four clusters themselves are clustered, thus guaranteeing a small overall L_2 distortion.

Note that we have not yet said anything about the detailed form of the noise model, not even whether it is a continuous function. It is determined entirely by the particular form of the distortion introduced by $y(h)$.

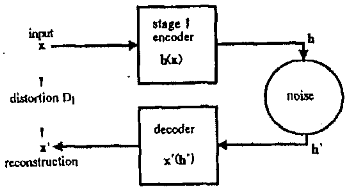

Fig. 2. Representation of (4) describing the L_2 distortion D_1 after encoding using $h(x)$ and modeling the distorting effect of $y(h)$ as a noise process that corrupts h into h', and then decoding using $x'(h')$.

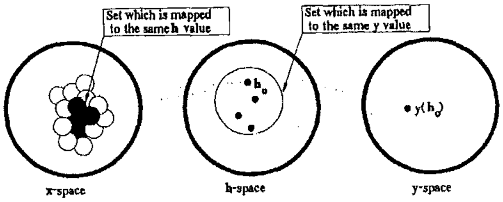
Set which is mapped to the same h value
Set which is mapped to the same y value
x-space
h-space
y-space

Fig. 3. Set theoretic representation of Fig. 2.

IV. A Modification of the LBG Algorithm

Now let us minimize D_1 with respect to $h(x)$ and $x'(h)$. The functional derivatives of D_1 with respect to $h(x)$ and $x'(h)$ are given by

$$\delta D_1/\delta h(x) = P(x) \int dn\, \pi(n) \cdot \partial\left[x'(h) - x\right]^2/\partial h\big|_{h = h(x) + n} \tag{5}$$

$$\delta D_1/\delta x'(h) = 2 \int dx\, P(x)\, \pi(h - h(x))\left[x'(h) - x\right]. \tag{6}$$

These two derivatives imply that the two necessary conditions A1 and A2 for minimizing D_1 must be modified to become:

B1) Given x, choose $h(x)$ to minimize $\int dn\, \pi(n)[x'(h(x) + n) - x]^2$.

B2) Given h, choose $x'(h)$ to satisfy

$$x'(h) = \left[\int dx\, P(x)\, \pi(h - h(x))x\right] \Big/ \left[\int dx\, P(x)\, \pi(h - h(x))\right]. \tag{7}$$

As a consistency check, if $\pi(n) = \delta(n)$, where $\delta(n)$ is a Dirac delta function (or the analogous Kronecker delta if we consider the discrete case), then the model reduces to the encoding/decoding depicted in Fig. 1 (with y replaced by h), and the above conditions reduce to the standard LBG vector quantization conditions, as expected.

It remains to determine a suitable form for the noise model $\pi(n)$, which depends on the distorting effect of the second stage of quantization $y(h)$. We need to be more specific about $y(h)$, so we shall now assume that $y(h)$ is a minimum L_2 distortion vector quantizer for h. Thus we

Exhibit AA Page 512

shall minimize D_2 with respect to $y(\boldsymbol{h})$ and $\boldsymbol{h}'(y)$, where D_2 is defined as

$$D_2 \equiv \int dh\, P_h(\boldsymbol{h})\left[\boldsymbol{h}'(y(\boldsymbol{h})) - \boldsymbol{h}\right]^2. \tag{8}$$

$P_h(\boldsymbol{h})$ is the PDF of the output $\boldsymbol{h}$ of the first stage of quantization. Given the value of D_2, and no further information about the distorting effect of $y(\boldsymbol{h})$, a zero mean Gaussian noise model with variance D_2 is then the appropriate maximum entropy choice for $\pi(\boldsymbol{n})$.

We may simplify B1 by Taylor expanding $\boldsymbol{x}'(\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{n})$

$$\boldsymbol{x}'(\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{n}) = \exp\left[\boldsymbol{n}.\nabla_h\right]\boldsymbol{x}'(\boldsymbol{h})\big|_{h=h(x)} \tag{9}$$

where ∇_h is the vector differentiation operator with respect to $\boldsymbol{h}$. This yields to second order

$$\int dn\, \pi(\boldsymbol{n})\left[\boldsymbol{x}'(\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{n}) - \boldsymbol{x}\right]^2 \simeq \left[1 + (1/2)D_2\nabla_h^2\right]\left[\boldsymbol{x}'(\boldsymbol{h}) - \boldsymbol{x}\right]^2\Big|_{h=h(x)} \tag{10}$$

where the statistics of the maximum entropy zero mean Gaussian noise model for $\pi(\boldsymbol{n})$ are

$$\int dn\, \pi(\boldsymbol{n}) = 1 \qquad \int dn\, \pi(\boldsymbol{n})n_i = 0$$

$$\int dn\, \pi(\boldsymbol{n})n_i n_j = D_2\delta_{ij}. \tag{11}$$

The first term on the right-hand side of (10) is the conventional distortion, and the second (curvature) term arises from the output noise model $\pi(\boldsymbol{n})$. We shall henceforth assume that the curvature term is small (i.e., $\mathcal{O}(D_2\nabla_h^2) << 1$, effectively), so that B1 may be approximated by A1 (with y replaced by $\boldsymbol{h}(\boldsymbol{x})$). This reduces B1 to a nearest neighbor encoding rule, as before.

B2 cannot be simplified in an analogous fashion to B1. However, from (6) we may derive a stochastic gradient descent algorithm for realizing B2. Thus we draw $\boldsymbol{x}$ samples at random from the training set using the $\int dx\, P(\boldsymbol{x})$ factor, and gradient descent of D_1 is then achieved by updates of the following form where $\epsilon > 0$, $\boldsymbol{x}$ is drawn from the probability density function $P(\boldsymbol{x})$, and $\boldsymbol{h}(\boldsymbol{x})$ is the nearest neighbor encoding approximation to B1:

$$\boldsymbol{x}'(\boldsymbol{h}') \rightarrow \boldsymbol{x}'(\boldsymbol{h}') + \epsilon\pi(\boldsymbol{h}' - \boldsymbol{h}(\boldsymbol{x}))\left[\boldsymbol{x} - \boldsymbol{x}'(\boldsymbol{h}')\right]. \tag{12}$$

This update prescription is applied to all $\boldsymbol{h}'$ for which $\pi(\boldsymbol{h}' - \boldsymbol{h}(\boldsymbol{x})) > 0$. This update scheme is identical to the one that Kohonen has proposed for obtaining topographic mappings [4], and it leads to a decoding operation $\boldsymbol{x}'(\boldsymbol{h})$ that is a smoothly varying function of $\boldsymbol{h}$, although the encoding operation $\boldsymbol{h}(\boldsymbol{x})$ is not guaranteed to be a smooth function of $\boldsymbol{x}$ when dim $(\boldsymbol{h}) <$ dim $(\boldsymbol{x})$. Note that we have had to make an approximation (where we retain only the leading term in (10)) in order to obtain the correspondence between a minimum L_2 distortion vector quantizer (with added code noise) and Kohonen's topographic mappings.

Our noise model does not make specific predictions about the dependence of $\pi(\boldsymbol{n})$ on the number of updates that have elapsed. However, D_1 has multiple local minima which makes it difficult to locate the global minimum by using the gradient descent algorithm specified in (12). The randomness of sampling from the training set makes unpredictable the order in which updates are performed according to (12), which alleviates the local minimum problem somewhat. However, a broader $\pi(\boldsymbol{n})$ gives rise to a D_1 which has a softer dependence on $\boldsymbol{h}(\boldsymbol{x})$ and $\boldsymbol{x}'(\boldsymbol{h})$, so better convergence to the global minimum of D_1 is obtained by starting with a broad $\pi(\boldsymbol{n})$, and then progressively reducing the width of $\pi(\boldsymbol{n})$ as the updates proceed until the required form is obtained [4]. This is equivalent to a renormalized scheme in which $\pi(\boldsymbol{n})$ is fixed while the underlying space is progressively dilated as the updates proceed [5]. The renormalized scheme gives much faster convergence than Kohonen's original scheme.

A similar approach to that presented in this paper has been presented elsewhere [6]. They minimize a quadratic potential which is analogous to our distortion measure D_1 in (4). However, our approach provides a more detailed interpretation of the neighborhood updating aspect of the training algorithm.

V. Extension to a Multilayer Vector Quantizer

We shall now use the robust hidden layer principle to state the form of (4) that applies to an adjacent pair of layers in a multilayer vector quantizer. Thus define a distortion $D(m)$ for encoding/decoding layer m of a multilayer vector quantizer

$$D(m) \equiv \int dx_m\, P_m(\boldsymbol{x}_m) \int dn\, \pi_m(\boldsymbol{n}) \cdot \left[\boldsymbol{x}_m'(\boldsymbol{x}_{m+1}(\boldsymbol{x}_m) + \boldsymbol{n}) - \boldsymbol{x}_m\right]^2 \tag{13}$$

where $\boldsymbol{x}_m$ is a vector representing the state of layer m of the quantizer, $\boldsymbol{x}_{m+1}(\boldsymbol{x}_m)$ is the encoding operation used to obtain the state of layer $m + 1$ from layer m, $\boldsymbol{x}_m'(\boldsymbol{x}_{m+1})$ is the decoding operation used to obtain the state of layer m from layer $m + 1$, and $\pi_m(\boldsymbol{n})$ is a noise process that models the distortion due to all subsequent stages of the multilayer vector quantizer. The input probability density function $P_m(\boldsymbol{x}_m)$ is obtained as shown where we now write $\boldsymbol{x}_1$ instead of $\boldsymbol{x}$ for the original input vector:

$$P_m(\boldsymbol{x}_m) \equiv \int dx_1\, dx_2 \cdots dx_{m-1}\, P_1(\boldsymbol{x}_1) \cdot \delta(\boldsymbol{x}_2 - \boldsymbol{x}_2(\boldsymbol{x}_1)) \cdots \delta(\boldsymbol{x}_m - \boldsymbol{x}_m(\boldsymbol{x}_{m-1})). \tag{14}$$

In practice, samples $\boldsymbol{x}_m$ from $P_m(\boldsymbol{x}_m)$ may easily be generated by propagating samples $\boldsymbol{x}_1$ from $P_1(\boldsymbol{x}_1)$ through the layers of the network until layer m is reached. Thus the output of layer $m - 1$ is used as the training set for determining the optimum encoding/decoding operations on layer m.

Exhibit AA Page 513

The advantage of modeling the distorting effect of all subsequent layers of the multilayer vector quantizer as a noise process is that the encoding/decoding operations associated with layer m can be optimized (almost) independently of the other layers' encoding/decoding operations. Coupling between these optimizations is achieved via the distortion models $\pi_m(\boldsymbol{n})$. Although this will not lead to an optimal overall encoding/decoding scheme, it is computationally much cheaper than simultaneously optimizing all of the coupled layers. Finally, when the multilayer vector quantizer has been trained, and we wish to decode $\boldsymbol{x}_m$ to obtain a reconstruction $\boldsymbol{x}_{1,\text{rec}}(\boldsymbol{x}_m)$ of the input vector $\boldsymbol{x}_1$, we should use the following formula which is the centroid of the set of input vectors that map to $\boldsymbol{x}_m$:

$$\boldsymbol{x}_{1,\text{rec}}(\boldsymbol{x}_m) = \left[\int d\boldsymbol{x}_1 \cdots d\boldsymbol{x}_{m-1}\, P_1(\boldsymbol{x}_1)\, \delta(\boldsymbol{x}_2 - \boldsymbol{x}_2(\boldsymbol{x}_1)) \cdots \delta(\boldsymbol{x}_m - \boldsymbol{x}_m(\boldsymbol{x}_{m-1}))\boldsymbol{x}_1 \right] \Big/ P_m(\boldsymbol{x}_m). \quad (15)$$

VI. Further Extensions

All of the results in this paper can be reexpressed using an arbitrary (non-L_2) distortion measure. The choice of a preferred form for $d(\boldsymbol{x}, \boldsymbol{x}')$ depends on whatever structure in $\boldsymbol{x}$ one deems to be important.

The LBG algorithm is the batch training version of Kohonen's vector quantization algorithm with zero neighborhood size. It is also possible to use (5) and (6) to derive a batch training version of Kohonen's topographic mapping algorithm.

We have presented elsewhere some applications of 2-stage and multistage vector quantization to low-level image processing [5], cluster decomposition of PDF's [7], image compression [8], and time series analysis [9].

A nice example of the use of the robust hidden layer principle has been reported in [10]. The problem studied there was the design of a codebook containing 8-b codewords that needed to be robust with respect to bit errors. $\pi(\boldsymbol{n})$ must therefore describe Hamming noise, which induces an 8-dimensional Hamming cube topology on the codebook. A worthwhile reduction of 3 dB in the reconstruction distortion was obtained by topographically organizing the codebook in this fashion.

VII. Summary and Conclusions

We have shown how topographic mappings arise naturally in a 2-stage vector quantizer whose output is made robust with respect to various types of distortion. The choice of neighborhood used at the output of the first stage is determined solely by the additional distortion introduced by the second stage of the quantizer. This principle generalizes directly to multistage quantizers. The advantage of our approach is that (variants of) Kohonen's self-organizing neural network may be derived from first principles.

We have encapsulated this robustness argument in the robust hidden layer principle: train your network in the presence of all anticipated distortions to its internal representations. The network will then do its best to choose internal representations that encode information in such a way that it is robust with respect to being damaged by the distortions.

A hierarchical multistage quantizer constructed along these lines is an adaptive pyramid processor in which successively cruder quantizations (or codings) of the input are extracted. Current research indicates that such neural networks show much promise [9].

References

[1] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inform. Theory*, vol. 28, pp. 129–137, 1982.

[2] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inform. Theory*, vol. 6, pp. 7–12, 1960.

[3] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95, 1980.

[4] T. Kohonen, *Self-Organization and Associative Memory.* New York: Springer, 1984.

[5] S. P. Luttrell, "Self-organizing multilayer topographic mappings," in *Proc. 2nd IEEE Int. Conf. Neural Networks*, vol. 1 (San Diego, CA), 1988, pp. 93–100.

[6] H. Ritter and K. Schulten, "Kohonen's self-organizing maps: Exploring their computational capabilities," in *Proc. 2nd IEEE Int. Conf. Neural Networks*, vol. 1 (San Diego, CA), 1988, pp. 109–116.

[7] S. P. Luttrell, "The use of Bayesian and entropic methods in neural network theory," in *Maximum Entropy and Bayesian Methods*, J. Skilling, Ed. Kluwer, 1989, pp. 363–370.

[8] S. P. Luttrell, "Image compression using a multilayer neural network," *Patt. Recog. Lett.*, vol. 10, pp. 1–7, 1989.

[9] S. P. Luttrell, "Hierarchical vector quantization," *Proc. Inst. Elec. Eng.*, vol. 136, pt. 1, no. 6, pp. 405–413.

[10] D. S. Bradburn, "Reducing transmission error effects using a self-organizing network," in *Proc. 3rd IEEE Int. Joint Conf. Neural Networks*, vol. 2 (Washington DC), 1989, pp. 531–537.

Exhibit AA Page 514